Representation of Strategic Choices in Structural Modeling[1]

Steven E. Salata[2] and Clive L. Dym[3]
Department of Civil Engineering
University of Massachusetts
Amherst, MA 01003

1 This work was supported by Grant No. 88-0006 from the United States Air Force Office of Scientific Research.
2 Research Assistant.
3 Professor of Civil Engineering.

(1c)

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release, distribution unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR-90 0882
6a. NAME OF PERFORMING ORGANIZATION University of Massachusetts Department of Civil Engineering 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research
6c. ADDRESS (City, State, and ZIP Code) Amherst, MA 01003	7b. ADDRESS (City, State, and ZIP Code) AFOSR/NA Bolling Air Force Base, DC 20332-6448
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Air Force Office of Scientific Research 8b. OFFICE SYMBOL (If applicable) NA	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR 88-0006
8c. ADDRESS (City, State, and ZIP Code) AFOSR/NA Bolling Air Force Base, DC 20332-6448	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.
61102F	2302F	B1	

11. TITLE (Include Security Classification)

Representation of Strategic Choices in Structural Modeling (U)

12. PERSONAL AUTHOR(S)

Steven E. Salata and Clive L. Dym

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final Technical	FROM 10/88 TO 8/89	(1990, June 29)	58

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Knowledge-Based Systems, Expert Systems, Structural Mechanics

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Structural modeling requires the construction of an appropriate mathematical description to describe the behavior of a physical object. Because conflicting goals and uncertainty permeate the process of structural modeling, structural model derivation is a complicated process. Therefore, to effectively and efficiently model any structure, one must have a method for planning actions, for proceeding in the face of uncertain information, and for dealing with uncertainty.

This paper presents a method for representing structural modeling as a strategic process involving decision-making in an environment where the results of any decision may not be known with complete certainty. Specifically, an existing task-level architecture developed to manage uncertainty in treating cardiac disease is adapted to embody the strategic knowledge of a structural engineer in formulating and solving structural problems. The MUMS system focuses on plates as the structure of interest in its pilot implementation. Since the system is concerned with structural modeling at the strategic level (as opposed to a detailed design, for instance), the ideas presented are applicable to modeling any structure.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☒ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Anthony Amos/Dr. Spencer Wu	22b. TELEPHONE (Include Area Code) (202) 767-6962 22c. OFFICE SYMBOL NA

DD Form 1473, JUN 86 — Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Unclassified

Table of Contents

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 The Structural Modeling Problem

Structural modeling is the process of forming a mathematical representation of a physical structure which will describe its behavior and/or performance. A major concern in structural modeling is the proper choice of tools to achieve stated objectives. Problem statements in structural analysis and design are usually made in fairly abstract ways, for example, in terms of high-level descriptions of the object being studied and the calculations being planned. It is up to the modeler to: refine this high-level description to appropriate levels of detail; choose and exercise one or more modeling tools; and interpret and assess the results produced. The structural modeling paradigm and it's component steps are illustrated in Figure 1.1. Using the data from the physical structure, the structure's physical response is found by evaluating both numeric and analytic models. The *process* of generating structural response from structure data is called structural modeling.

A more abstract view of the modeling problem suggests that there are other considerations that enter into the choice and exercise of structural models. Some of these evolve from considering design as well as analysis. Other considerations could be viewed in the context of linking the geometrical, functional, and behavioral aspects of a structure in order

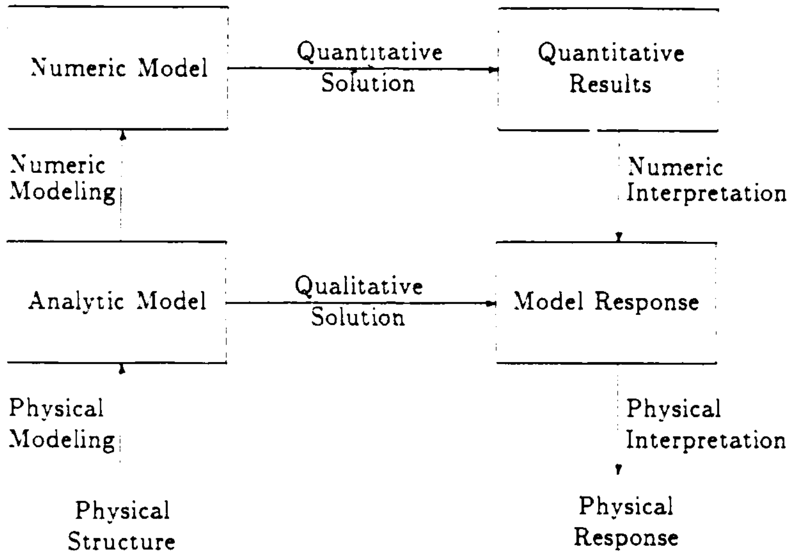


Figure 1.1 The Structural Modeling Problem (after [14])

that appropriate models are developed at each level of a design process [14].

Structural modeling is further complicated by the fact that knowledge of the object to be modeled is incomplete; there may be conflicting or alternative goals; the utility of actions may be influenced by other actions; there may be tradeoffs or constraints on resources; and actions may produce unforseen consequences in the state of the modeling problem. As a result, structural modeling tends to be a heuristic task, dependent on specific modeling problems and situations.

1.2 Approaches to Structural Modeling

Attempts to address the difficulties described above have been made. Because of the heuristic nature of the structural modeling process, algorithmic solutions have not been successful. Developments in artificial intelligence (AI) and knowledge-based expert systems (KBES),

however, have allowed reasoning on a level which is sufficiently abstract to adequately represent the structural modeling problem.

SACON, an early knowledge-based system, was built in part with an eye toward some of these issues [1]. An application of the diagnostic EMYCIN environment, the SACON system was based on a system developed to advise physicians on the diagnosis and treatment of infectious diseases [13]. SACON was concerned with capturing the knowledge of structural engineering experts about the use of the MARC FEM package. In particular, it was intended to encapsulate the pre-processing knowledge needed to choose an appropriate analysis class, identify and apply the rules pertaining to the controlling behavior of a structure, and suggest the appropriate mathematical model (implemented within the FEM) for performing the relevant calculations. An implicit boundary condition for the entire SACON project was that an FEM package was the vehicle for whatever analysis was called for [1].

Another project, the Buckling Expert, had similar aims, but extended its coverage to incorporate suggestions for the user on interpretation of the results (post-processing) and on possible re-design of the structure to achieve better behavior using multiple analysis codes [15]. The Buckling Expert was a rule-based system designed to act as a expert consultant during preliminary design of a shell structure. The system did involve the integration of analysis codes with an expert in the process of building a structural model. However, the analysis codes were only used at very specific, well-defined points in the model's history.

1.3 Motivation for the MUMS Research

The focus of the present thesis is related to the work of Turkiyyah and Fenves [14], but it pertains to a somewhat different abstraction of the structural modeling and analysis process. The emphasis herein is on the strategic choices in structural modeling, i.e., those

concerned with the choice of method in the context of issues of time, purpose, cost, and so on. Thus, we are concerned with developing a modeling plan that would allow the user to make choices in several dimensions, including the following:

1. What is the purpose of the proposed calculations?
2. What kind of information is sought, and with what granularity?
3. Does the information have economic value?
4. Are there issues of timeliness that affect the choice of model?
5. What kinds of information are available as input to the model?
6. Can the functional, behavioral, and spatial aspects of a structural system be represented and integrated for this strategic task?
7. What engineering tools and methods are available (e.g., FEM codes, analytical formulations, handbooks, back-of-the-envelope calculations, experiments)?
8. How are each of the tools and methods evaluated with respect to the decision dimensions outlined the above seven questions?

The intent of this research is to provide computational support for — and in the process make more explicit — the linking of numerical models with the intent behind their use in engineering analysis and design. Of course, what makes this linkage hard to accomplish is that the representations of the functional and behavioral aspects of a structure are likely to be considerably different in expression. Thus, even if the syntax is similar (i.e., the underlying geometrical representation of a building whether expressed in a

CADD drawing or a FEM mesh), the semantics will not be. And it is the delineation and expression of the differences in meaning that make this hard.

We are also concerned with knowledge acquisition because we are aware that any such tool is likely to be adopted by the engineering community only if it can be customized by the user. That is, every design group has its own culture and its own approach to analysis-in-design. To the extent that flexibility can be offered for the integration of local culture, the chances that this technology will be adopted improve. Thus, one of the considerations of the choice of architecture in this project is the ability to represent such knowledge at the task-level, that is, at a strategic level rather than at the implementation-level of primitives such as rules, frames, and so on that are usually used in knowledge-based systems. It is not that such primitives are not used, but that the knowledge acquisition aspects are such that the user can focus on the strategic domain issues.

A few final contextual notes. This project may also be seen in the larger context of using knowledge-based systems to do better engineering by integrating them with other, numerically-based programs that are in widespread use. The coupling of symbolic and numeric computing is increasingly of interest and reports of work along these lines are beginning to appear [9]. One example that might be of interest is the design of a knowledge-based system for architectural code-checking, the LSC Advisor, to be used within an architectural CADD system [5]. It is also worth noting that some of the strategic choices to be modeled in this work, although made in a static environment, have parallels in decisions that in some circumstances would be made more dynamically. Thus, recent work on real-time decision problem solving could perhaps also be of interest [10,3].

1.4 Strategic Knowledge in Structural Modeling

It turns out to be useful to classify knowledge as either substantive knowledge or strategic knowledge. We closely follow Gruber's distinction between the two types of knowledge [7]. Gruber identifies *substantive* knowledge as knowledge about "what is believed about the world" and *strategic* knowledge as "knowledge used to decide what course of actions to take when there are conflicting criteria to satisfy and the precise effects of actions cannot be known in advance". Another way of framing the distinction between the two types of knowledge is to differentiate between the "rules of the game" (substantive knowledge), on the one hand, and "how to play the game" (strategic knowledge), on the other.

One example of the use of substantive knowledge in structural modeling is the following. In a beam bending problem, to calculate bending stress at a point, one must first find the moment at that point, the distance of the neutral axis to the outer fiber, and the moment of inertia of the cross section. In other words, substantive knowledge is knowledge which is widely accepted and very specific domain knowledge. Another example of substantive knowledge knowledge is the statement that "if the structure is a beam or a plate and there is an in-plane load, then buckling is possible." One example of strategic knowledge in the structural modeling domain is the statement that "if the model parameters are not certain, then start with a model based on an analytical formula." While substantive knowledge is necessary for generating a specific structural model, strategic knowledge is essential for efficiently managing the *process* of structural modeling.

CHAPTER 2

The MU Architecture

To understand the MUMS system, it is first necessary to understand the MU (MU is an acronym for "managing uncertainty") architecture upon which MUMS was built. The MU system is a programming environment for knowledge systems developed in the Experimental Knowledge Systems Laboratory at the University of Massachusetts [3].

2.1 Overview

The MU environment is a task-level architecture developed for reasoning with incomplete or uncertain knowledge. It evolved from the underlying ideas in a program called MUM (Managing Uncertainty in Medicine) which planned sequences of actions for the diagnosis of chest and abdominal pain [3]. The goal of the MUM research was to create a system to manage uncertainty in the diagnosis of chest pain. Emphasis was placed on studying the *process* of the diagnostic sequence of questions and tests the physician conducts. The MUM research project resulted in the creation of the MU system which has the following characteristics:

1. The MU system assists in transforming strategic knowledge in the *acquirable* form (as used by the expert) to the *operational* form (as used by an expert system).

2. The MU system is an example of how strategic knowledge will produce efficient solutions to tasks in which uncertainty is a factor.

3. The MU system is a task-level architecture applicable to many fields in which experts use similar strategies to solve problems efficiently.

One noteworthy design characteristic of MU is its lack of a predetermined control strategy. The problem-solving strategies used in MU are defined in the control features in an application's domain-specific knowledge. This control knowledge is acquired from the expert and implemented by the knowledge engineer. In addition to the structural modeling domain, the MU architecture has been applied experimentally to the fields of plant pathology and fighting forest fires. The most important aspects of the MU architecture are described in Sections 2.2 through 2.5.

2.2 Features

Features incorporate the information or evidence used in planning a strategy, evaluating hypotheses, and making decisions. They are central to the operation of MU. For instance, in diagnosing chest pain, a doctor collects evidence to support or deny a hypothesis he or she might have. The evidence collected will depend on the features such as the reliability of a test, or the cost of obtaining that evidence. In this role, features are used to guide the diagnosis and arrive at a conclusion efficiently.

By observing expert problem-solving, it is apparent that experts make extensive use of features — at various degrees of conspicuousness — in performing diagnostic tasks. MU leaves to the knowledge engineer the task of identifying, defining, and making these features operational in the MU knowledge base. There are four classes of features are which are identified in MU:

- *Static features* are extracted from knowledge acquisition sessions with the expert and, as the name suggests, do not change their values as the knowledge base (KB) is run. The time taken to perform a specific test is an example of a static feature.
- *Datum features* are features which are found by prompting the user or performing actions. An example of a datum feature is the result of specific test.
- *Dynamic features* are computed by evaluating features. The expert specifies how the dynamic feature is affected. For example, a *degree of belief* in a hypothesis changes with changes in its evidence
- A *focus feature* value is used to concentrate on or divert attention away from certain actions. An example is the *differential* feature which contains the hypotheses or tests which require the greatest attention.

In MU, instances of a feature are associated with evidence or hypotheses by means of a combination function in a slot in the evidence or hypothesis frame. We define these functions in the next section.

2.3 Combination Functions

An important aspect of the MU architecture involves combining evidence gathered during the execution of the KB. This is the task of *combination functions*, which are essentially IF-THEN rules specified by the expert. Unlike some knowledge systems, MU uses only local combination functions; that is, a specific frame may include a combination function whose value is used only within that frame and is not directly propagated to any other node. However, combination functions in other nodes could access that value (if needed) to calculate their own values. Local combination functions have two important benefits: (1)

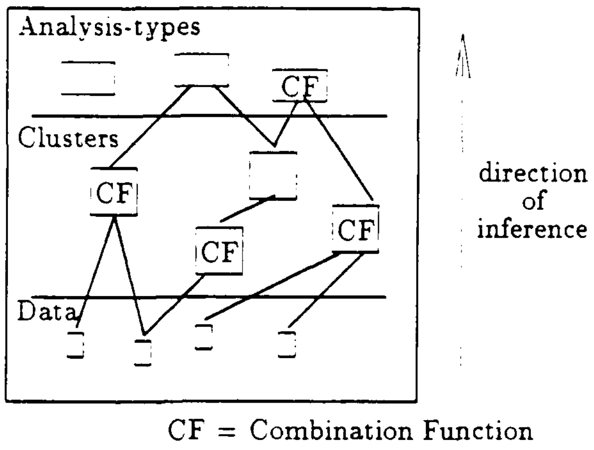


Figure 1.1 The inference network in MU/MUMS

they are similar to the way diagnosticians actually use information [?], and (2) they are easy to acquire and implement.

Combination functions serve two purposes in MU. First, they are the means by which information between nodes is shared and updated in the inference network (see next section). With each additional piece of evidence or data, the combination functions are run and the results are stored. The second purpose of the combination function is to provide links of causality which may be exploited for prospective views of actions. In other words, combination functions provide answers for "What if ...?" questions.

1.4 The Inference Network

The movement of data by the combination functions creates an inference network in MU (see Figure 1.1). The *inference network* is the means by which information is moved to make intermediate conclusions about the problem.

Conceptually, nodes lower down in the inference network "support" the nodes above them by providing data (or evidence) which is used by these higher nodes. Evidence is combined into significant groups (called *clusters*) by that node's combination function from data generated in the lower nodes. The data may be answers to questions posed to the user or the result of some test or analysis. These data are then combined in the clusters and further combined in the hypotheses nodes to arrive at a conclusion. In the inference network, the clusters and hypothesis are conclusions based on data that has been entered by the user or data that has been concluded from data or from other conclusions.

2.5 Strategy Rules

The MU architecture does not inherently provide a specific strategy. There are no domain-specific strategy rules to choose an action in a given situation. Rather, MU provides a general strategy rule control cycle which loops through the following rules until the problem is solved (i.e. a goal is found):

- *Focus rules* choose actions which are possible in a given state.
- *Filter rules* remove actions from consideration in a given state.
- *Preference rules* specify one set of actions over another.

In the focus $\Longrightarrow$filter $\Longrightarrow$prefer cycle, MU does not provide criteria for assessing how or why an action should be focused upon, filtered, or preferred. Ultimately, the rules are dependent upon the application. The rules are, however, closely tied to the *features* mentioned in

Section 2.2. An example from the MUM knowledge base is as follows.

```
 IF (= (current-goal) quick-diagnosis)
AND (in ?action (proposed-actions))
AND (> (cost ?action) low)
THEN (filter ?action)
```

This rule will filter actions which have a cost feature greater than "low" if the current goal is a quick diagnosis. The strategy rule cycle (focus on repeats until a goal is reached or until no actions are possible, e.g., until all actions have been filtered. For such an impasse, the Acquiring Strategic Knowledge (ASK) assistant [8] was developed.

CHAPTER 3

Acquiring Strategic Knowledge for Structural Modeling

3.1 Background

MUMS is a KBES designed to aid in modeling structural plate problems. The knowledge contained within the MUMS system can be divided into the two types mentioned in Chapter 2: strategic and substantive. The strategic knowledge is used as a "tour guide" to manage decision-making in the structural modeling process. The substantive knowledge – the knowledge about the physical world – is used to conclude facts about the structural model given some set of structural modeling data. While both types of knowledge are necessary in effective structural modeling, each type is acquired by different means. For instance, knowledge used for answering questions such as "How do I model a plate when the loading is uncertain?"; "How can I arrive at a model when I have limited time?"; or, "What type of model do I choose when I have both serviceability and strength requirements to consider?" is almost exclusively a product of experience derived over many years of structural modeling. Whereas, substantive knowledge (e.g., "Does the geometry and loading of the problem indicate a bending problem?", "Will the plate be subject to large vibrations?", and so on) may be derived from the structural literature or from the structural modeling expert.

To obtain the necessary strategic and substantive knowledge for the MUMS system, two sources were used: (1) interactive knowledge acquisition sessions with an expert structural modeler, and (2) the structural modeling literature. The major source of the strategic structural modeling knowledge for the MUMS system was the domain expert, Dr. Clive L. Dym of the University of Massachusetts Department of Civil Engineering. Many sample plate problems were generated for which Dr. Dym provided a structural model and analysis. The problems were structured so as to illuminate the issues involved in the structural modeling process (see Section 3.3). The remaining structural modeling knowledge to be used by MUMS was obtained from the literature. The knowledge from the structural modeling literature is almost always substantive knowledge. It will not yield information for directly deciding which "direction" a structural model should take. For example, the equation governing the deflection of a plate can be found in any book on the analysis of plates (e.g. [6]):

$$\nabla^4 w = \frac{q(x, y)}{D}$$

As far as the strategy in the structural modeling process is concerned, this is where the usefulness of the literature ends. It is up to the structural modeler to decide if the equation is *even* valid for the problem at hand; if other models or analyses can be constructed to yield adequate results with less effort; or if a more detailed analysis is prudent.

3.2 Identifying Knowledge to be Captured

One of the first tasks in developing a structural modeling system like MUMS is to identify the knowledge to be acquired. To do this, a representation of the structural modeling process was created. Such a representation must accurately reflect the actions and decisions of the structural modeling expert while at the same time must make explicit the knowledge to be

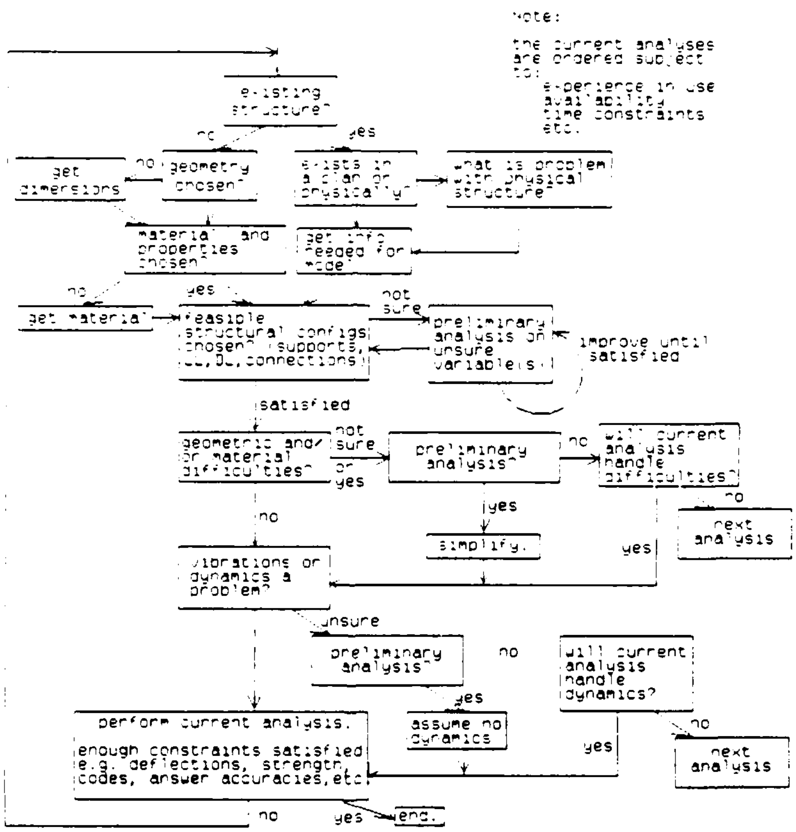


Figure 3.1 Structural Modeling Representation for Knowledge Acquisition

captured. Figure 3.1 shows the representation we developed and used for this study.

This figure illustrates the structural modeling process in general, although the only concern here is with the modeling process as it relates to plate structures. On closer inspection of Figure 3.1, it may be seen that elements of uncertainty may be present at certain points in the structural modeling process. Also, it may be noticed that this uncertainty may have an affect on later actions – even whether these later actions are performed or not. For example, one may not be sure of the values assigned to the modeling variables. Are the loads completely known? Just how certain are these loads? Is the structure's geometry adequately represented in the model? Should dynamics be included?, etc.

There may be uncertainty in in the results of performing certain modeling actions (e.g., "Has a preliminary analysis satisfied me that they are adequate for my needs?"). Even the modeling goals may be less than certain (e.g., is the structure's strength more important than the structure's serviceability?).

MUMS is an attempt to make explicit the issues involved in answering these questions. For any given box in Figure 3.1 one would like to know: (1) *why* the expert proceeded to that box on the path from problem description to complete structural model; (2) *what* features of the problem led to a decision; and (3) *how* the features affected the modeler's decision. The answers to these questions comprise the structural modeling strategy.

3.2.1 Eliciting Knowledge from the Expert

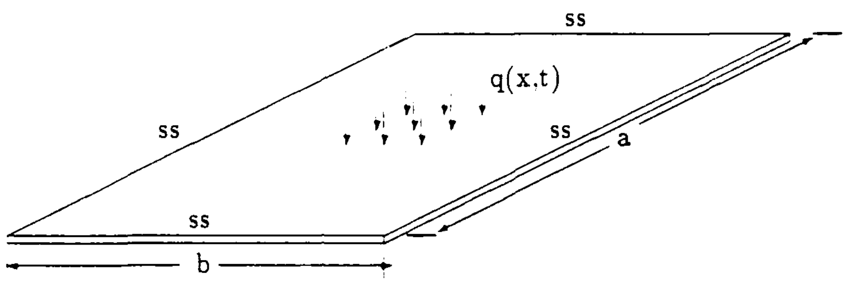
One of the main reasons for producing the structural modeling representation of Figure 3.1 was to identify the knowledge (strategic and substantive) needed for MUMS. Once the knowledge was identified, a means for obtaining and transforming the knowledge for use in MUMS was devised. This section describes the means for acquiring the knowledge for MUMS.

The task of transferring structural modeling knowledge from an expert to the MUMS system was accomplished using knowledge acquisition sessions with Dr. Clive Dym as the domain expert and Steven Salata as the knowledge engineer. The sessions consisted of interviews with the domain expert in which sample plate problems were presented to the expert and the steps he used to solve the problems were recorded. Because of the structural mechanics expertise of the domain expert, the sample problems and subsequent knowledge acquisition sessions concentrated on plate problems using analytical solutions.

The sample plate problems and knowledge acquisition session were structured so that knowledge acquisition about strategic decisions could be easily isolated and extracted from the session protocol. The domain expert was asked to *actually* solve the problems rather than to describe *how* the problems should be solved [11]. Gruber [7] has found that experts most easily convey their problem-solving strategy through justifications of their actions. He found that experts had difficulty in explaining their strategy but could easily give reasons for their actions. Thus, problems were designed to focus on the factors (which we will call "features" from now on) which affect the structural modeling process. The precise effect these features have on the process of structural modeling is of tantamount concern (see Section 5.1 for more on features and their uses in MUMS). Again, from the knowledge acquisition we are trying to discover what (and how) features led to a decision in the structural model. An abbreviated example problem and resulting knowledge acquisition session are presented to illustrate some of the knowledge acquisition ideas. In this example, **EX:** indicates comments made by the expert and **K_E:** identifies comments made by the knowledge engineer.

Problem:

A structural model is to be completed on the following simply supported structure, thickness h, a/b = 2, under a distributed load, *q(x,t) applied at the center:*



K_E: Here is a picture of the problem. You are to provide a structural model and analysis.

EX: What is the purpose of the analysis for this structure?

K_E: You are to perform a detailed design.

EX: What is the nature of the load?

K_E: It is a continuous function of time and space.

EX: If a detailed design is needed, long-term and short term responses will be computed.

EX: I will use $D\nabla^4 w + \rho h w = q(x, y, t)$ since the solution converges quickly, because of the problem's simple geometry, and because the plate is simply supported all around.

EX: What is the thickness of the plate?

K_E: You can assume that the thickness is small in comparison to a or b.

EX: I will now find the natural frequency of the plate.

Then I will find the deflections from biharmonic equation.

Then I will find the support forces, the stresses, and the strains resulting from deflections.

After the expert has completed the structural model, questions are asked by the knowledge engineer to clarify points which may be ambiguous or which may need further explanation.

K_E: Why did ask about the nature of the loading?

EX: The loading type is needed to find the period of interest which is different for shock, harmonic loading, step...

K_E: What would you do differently if the loading were a shock load?

EX: I would first find the approximate response (which is double the static response) to compute approximate magnitudes of stress, strains. From the approximate magnitudes of the stress and strains I can then decide if a linear analysis is appropriate.

K_E: Why did you choose the biharmonic equation?

EX: The solution is general and is easily solved for these loading and support conditions.

K_E: Why did you ask the thickness?

EX: If the thickness is on the same order of magnitude of the other two dimensions of the plate, then shear deformation may be significant.

This knowledge acquisition example shows how the strategic information was acquired. Questions such as : "Why did you do this task?", "Why is this important?", and "What if this fact were true?" provide clues to the structural modeler's strategy. From the knowledge acquisition sessions, we try to identify (1) the features used by structural modeler's, and (2) how the features are used by experts in formulating a structural modeling strategy.

However, a fact concerning modeling strategy is hidden in this example. It is the fact that the expert is an expert in using analytical solutions for structural mechanics problems. This fact is made evident in this sample problem and clearly influenced the analysis type chosen (and the modeling process in general). Most experts in structural modeling are experts in one — or at most a few — type of analysis which influence the strategy they will choose in creating the structural model. Thus, they will develop modeling strategies (consistent with the problem features) to take advantage of their analysis strengths.

3.2.2 Judging the Suitability of Knowledge

As the structural modeling knowledge is extracted from the knowledge acquisition session protocols, it must be judged for suitability and, where appropriate, inserted into the KB. Whereas the strategic knowledge of an expert in structural modeling may be idiosyncratic (as hinted above), it does work. If it did not, an expert in structural modeling would not be considered an expert. Judging the suitability of substantive knowledge, the the other hand, is much simpler. Substantive knowledge acquired during knowledge acquisition can often be verified from the literature.

Once the strategic and substantive knowledge is obtained, it must be put in an operational form to be used by the MUMS system. The issues involved in transforming and representing expert knowledge for use by MUMS are presented next.

CHAPTER 4

The MUMS Knowledge Base

4.1 Background and Overview

The details of the MUMS system are described in this chapter. As indicated in Chapter 2, MUMS is an application of the MU KB. MUMS (and MU) is implemented on a Texas Instruments Explorer^{TM} II workstation using the KEE^{TM} programming environment. KEE provides the AI programming constructs of which MUMS is based — frames, slots, and facets. Frames are knowledge structures used to group together a collection of attributes that a given object normally possesses [12]. The attributes and their associated information are stored in the frame's slots. These attributes are not necessarily constrained to physical attributes of the objects they describe. They may also contain procedures for obtaining their values. Finally, each slot of the frame has many facets which contain implementation details of the slot's allowable values, how the slot is displayed, etc. KEE^{TM} also allows specific frames to inherit attributes from a more general frame.

The structural modeling knowledge in the MUMS KB is embodied entirely in the features, the strategy rules, and the frames of the inference network. As structural modeling knowledge is acquired from the expert, it is evaluated and inserted into the MUMS system. This chapter describes the means for the evaluation and representation of structural

modeling knowledge in the MUMS KB.

4.2 Features

Section 2.2 introduced *features* as incorporating the information experts use to make decisions or, in effect, form a strategy. Section 3.2.1 explained that features are acquired from the expert. More specifically, identifying and judging the significance of a feature is usually accomplished during many knowledge acquisition sessions. During these sessions, particular attention is paid to the facts of the problem which produce actions in the modeling process (as opposed to those facts which yield conclusions, for example). Figure 4.1 shows the features section of the MUMS KB. The four types of features (data, dynamic, static, and system) dictated by the MU system are clearly visible. To the right of the four types of features are those features identified in the structural modeling process.

One of the observed features is the **expertise** of the modeler in exercising a particular structural analysis. Choosing a particular structural analysis is influenced (and complicated) by many features, including the level of expertise the modeler has gained in any one analysis type. For example, a modeler may have many years experience with a particular structure type using a particular finite element package and thus would have a preference for using the familiar analysis strategy on a problem which also seems familiar. The **expertise** feature frame in MUMS is the representation of this fact.

Notice in Figure 4.2 that **expertise** is a static feature. Static features will not change during the modeling process — the modeler's expertise is considered a constant. The **expertise** feature of a particular analysis type is assigned its value when MUMS is first executed. At this point, the user is requested to enter his or her expertise in the analysis types of which MUMS knows. In addition to a type, feature have a value. The **expertise**

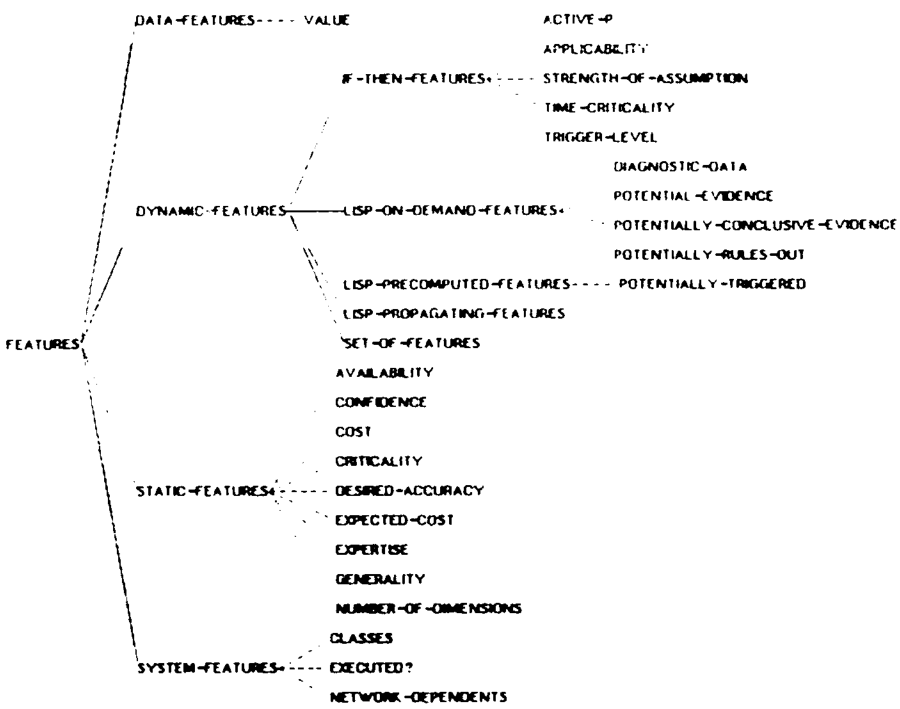


Figure 4.1 The features section of the MUMS KB

expertise

Feature-type: Static
Value-type: Ordinal
Value-range: (novice average expert)
Combination-function: local to an analysis
Value: to be inserted in the appropriate analysis type

Figure 4.2 The definition of the **expertise** feature

feature's value can be either *novice*, or *average*, or *expert*. From knowledge acquisition sessions, these three feature values of **expertise** were identified as being significant in affecting how an analysis is chosen. Finally, the "Combination-function" and "value" slots are included to indicate that this feature's value is not used locally in the **expertise** frame. Rather, the value of **expertise** is a feature of an analysis and is stored with that analysis. Once the value is initially set by the user, it will be used by the analysis types as needed.

Another feature type of the MUMS system is the dynamic feature. Unlike the static features whose values are set *before* the modeling process is begun, the dynamic feature derives its value from the evaluation of other features. Its value is computed during the modeling process. An example is the **trigger-level** feature of Figure 4.3.

trigger-level

Feature-type: Dynamic
Value-type: Ordinal
Value-range: (triggered not-triggered)
Combination-function: local to an analysis
Value: to be inserted in the appropriate analysis type

Figure 4.3 The definition of the **trigger-level** feature

While the **trigger-level** feature appears to be very similar to the **expertise** feature, its value is computed quite differently. To see how the value of a dynamic feature is found, one must look at a node in the inference network that uses this feature. An example is the **buckling** node which has the following combination function:

```
  IF value Of in-plane-load
  IS known
 AND value OR in-plane-load
  IS NOT none
THEN trigger-level OF buckling IS triggered
```

This rule translates to the statement that, "If there are in-plane loads on the plate, then there is a possibility of buckling." This combination function brings the suggestion of buckling into the structural model. Just how the buckling is dealt with later in the model is not specified since buckling might be handled differently depending upon how the modeling progresses.

The other two types of features in MUMS are system and data. System features are used by the MUMS system to keep the inference network (and thus the structural model) updated. Data features are simply the **values** of the model data. An example of a data feature is the value of the structure's material.

Features play their main role in combination functions within nodes in the inference. Section 4.4 illustrates more uses of features in combination functions. See Appendix A for a listing of the complete listing of features we have identified for structural modeling.

4.3 Strategy Rules

The MU system from which MUMS is based does *not* provide a pre-defined strategy for solving problems and, in that context, MU applications are not constrained to a particular strategy. However, MU does supply the means for easily creating a strategy. This is done by allowing the knowledge engineer (or the accompanying ASK knowledge acquisition program) the ability to add strategy rules to provide the necessary strategic control of the system. The strategy rule control cycle MU provides is shown in Figure 4.4.

To illustrate how the control cycle operates and to clarify the function of each strategy rule type, consider the following example from the MUMS KB. When the MUMS system is first begun, the strategy rule cycle is invoked and the first step (Run focus rules) is taken. The function of the focus rules is to indicate which actions are possible (from all actions) when the current conditions of the structural model are considered. All focus rules are tested and any focus rule which is applicable in that given situation is run. In this example, Focus rule 2 is the first rule to be run:

```
Focus rule 2: Ask Identifying Questions
If:
   (IS (DIFFERENTIAL) :EMPTY)
   (IN ?ACTION
        (MEMBERS-OF INITIAL-QUESTIONS))
Then:
   (PROPOSE ?ACTION COMPLETE-MODEL)
```

Since there is no active hypothesis (e.g. there are no structural analyses which we are considering), both Focus rules 2 and 3 propose questioning the user for input. These actions are then passed to the filter rules which remove some of the questions from consideration

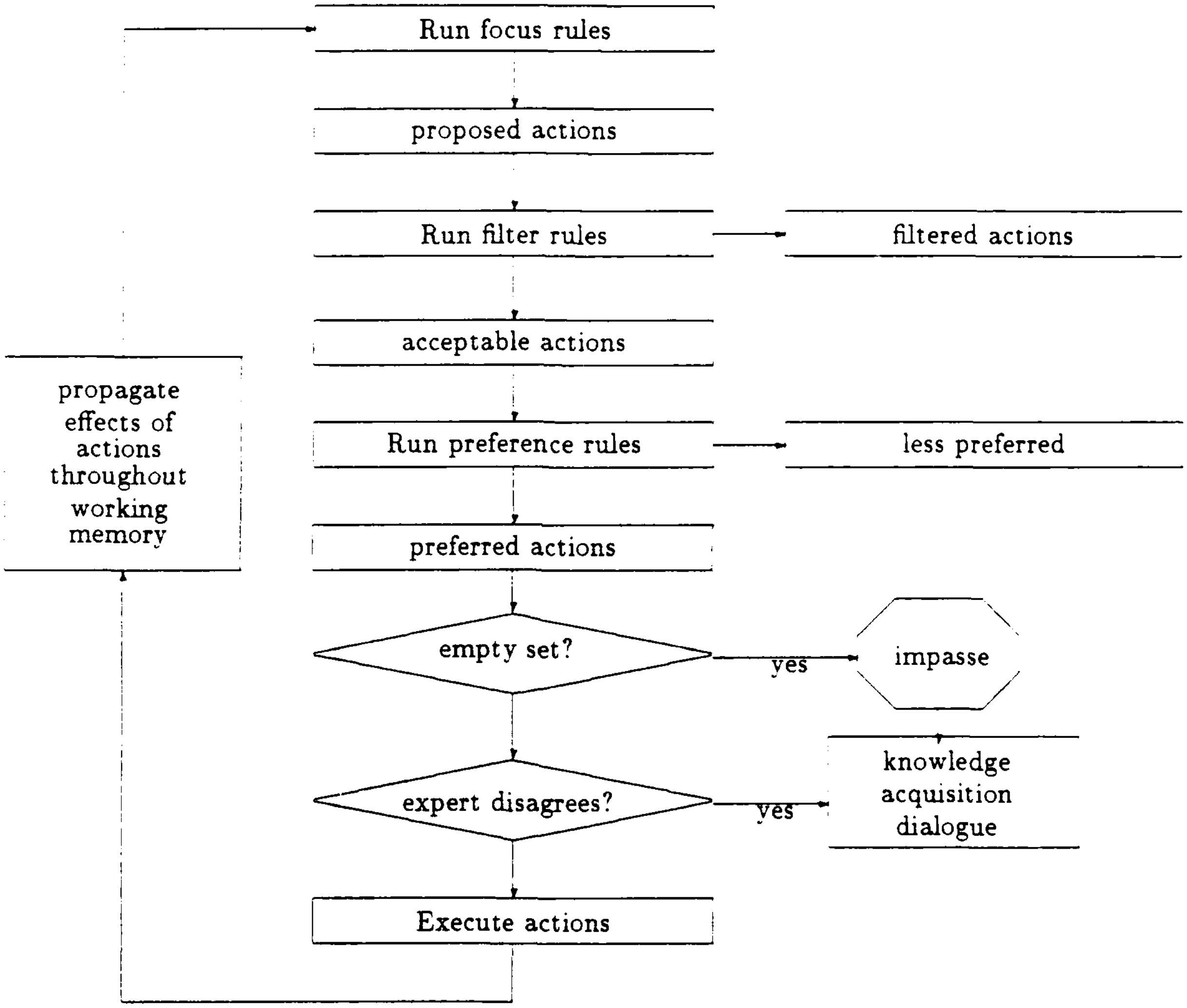


Figure 4.4 The strategy rule control cycle in MU/MUMS [8]

— because those questions are inapplicable — using Filter rule 2:

```
Filter rule 2: Filter Inapplicable Questions
If:
      (IS (APPLICABILITY ?ACTION)
          INAPPLICABLE)
Then:
      (FILTER ?ACTION needs prerequisites)
```

The questions which haven't been removed from consideration are then passed to the preference rules. The *preference rules* will act on the remaining questions to choose the "best" one to ask (the best action). The "best" action is the action which will accomplish "the most" given the present state of the model. This may seem vague, but the preferred actions may depend on many (possibly conflicting) criteria. For a complete listing of MUMS strategy rules see Appendix B.

A major reason for choosing to study the application of the MU architecture to structural modeling was the similarity observed in the two fields of prospective medical diagnosis and structural modeling. For example, both prospective medical diagnosis and structural modeling are concerned with choosing actions with the minimum cost and maximum safety, both tasks involve acting under uncertain conditions, and in terms of the MU system, both tasks focus on the management of a process as opposed to just obtaining the result of a process.

4.4 The Inference Net

Most of the knowledge in MUMS (and MU) is resides in the system's inference network. Recall from Section 2.4 that the *inference network* consists of nodes containing some form

of knowledge and the links between the nodes. These links dictate the effects the two nodes at the ends of the link will have on each other. It may be useful to visualize the modeling knowledge moving from one end of the inference network to the other as a structural model evolves. When a structural modeling problem is first undertaken, a modeler has a set of "givens" and/or assumptions about the problem. In the inference network, these are called *data*. And, as the model progresses, the modeler makes certain intermediate inferences about the model by combining the data he or she was given or assumed. In the inference network these combinations of data are called *clusters*, while the rules for combining the data are called *combination functions*. Depending upon the complexity of the model, a structural modeler may combine and re-combine data many times to create additional inferences (clusters). The ultimate goal is for an inference (or many inferences) to point to or suggest the correct analysis type for the given features of the problem. In the MUM system for the diagnosis of chest pain, this is analogous to diagnosing the correct ailment. As with chest pain diagnosis, the structural modeling process in MUMS should not stop there. Once a structural analysis is performed, new information may become available which may affect the objectives which the model is intended to achieve. Until all objectives are satisfied, the new information will be included in the inference network with the aim of performing the correct (and possibly different) structural analyses.

In the MUMS research, a major task has been the identification of the significant groupings of data used by the expert in arriving at intermediate inferences. Along with the groupings of significant structural knowledge, there is an interest knowing *how* the knowledge was combined. An example from the MUMS KB will illustrate the knowledge being sought. Figure 4.5 shows portions of clusters of modeling data and how the data are combined to form new conclusions. Each represents a grouping of evidence leading (eventually) to an analysis in structural modeling.

```
cluster      : linear-model
combination
function     : IF confirmed plate-structure OR confirmed beam-structure
                  AND confirmed deflection < h/L
                  AND confirmed material = steel
               THEN confirmed


cluster      : linear-beam-theory
combination
function     : IF confirmed pure-bending
                  AND confirmed beam-structure
                  AND confirmed linear-model
               THEN confirmed


cluster      : membrane-structure
combination
function     : IF confirmed linear-beam-theory
                  AND confirmed top-fiber-stress >> bottom-fiber-stress
               THEN confirmed
```

Figure 4.5 Some clusters for structural modeling

```
analysis     : flexure-formula
triggered-by: confirmed linear-beam-theory
combination
function     : IF confirmed objective = find-stress
                  AND confirmed beam-structure
               THEN confirmed
               IF confirmed objective = find-stress
                  AND confirmed membrane-structure
               THEN disconfirmed
               IF confirmed objective = find-stress
                  AND confirmed plate-structure
               THEN supported
```

Figure 4.6 Part of the analysis frame for **flexure-formula**

In a structural modeling session, the general strategy is to first gather data for the model. As the data is acquired, it is combined in clusters such as those in Figure 4.5. In this example, the **linear-beam-theory** cluster uses the values of **plate-structure**, **beam-structure**, **deflection**, and **material** to get its "confirmed" value. When **linear-beam-theory** is confirmed, it can then be used to support other conclusions made later in the structural model. The **flexure-formula** is an example of another node in the inference network that uses the value of the **linear-beam-theory** cluster:

Just as data is combined in clusters, clusters are combined to lead to a structural analysis cluster. In this case, the **flexure-formula** is shown. Here, the cluster **linear-beam-theory** is used to trigger the hypothesis that the flexure formula is the appropriate structural analysis to be performed on the current model. A *trigger* in MUMS is a defined feature which immediately activates a hypothesis (the **flexure-formula**, in this example) when some piece of data is found (here, the **linear-beam-theory** cluster). Thus, the flexure formula analysis will be suggested here when a model based on the linear beam theory is concluded. The flexure formula will *not* always be the proper analysis approach when a linear beam model is concluded, however. For example, Figure 4.6 shows that when the

current modeling objective is to find an internal stress and the structure has been concluded to be a plate structure (concluded from another cluster), the flexure formula is "supported" but not "confirmed". This means that the flexure formula could be appropriate but that other data are needed to confirm its use. The other data could affect whether the flexure formula has already been applied to model or whether the application of the flexure formula will provide any new (and needed) data for the model.

CHAPTER 5

MUMS Plate Problem

An example of the MUMS system in operation is now given. In this example modeling session, MUMS is demonstrated on the user-level, i.e., as it would appear to the user running the system. At the user's level, MUMS appears to be following the general modeling strategy shown in Figure 5.1. When the system is first started, its focus is on gathering data for the structural model. Most of the data for the structural model is requested from the user. As modeling data is entered into the system, it is propagated through the inference network, providing intermediate modeling conclusions. Eventually, the modeling data lead to the choice of an analysis. Performing the analysis, in turn, creates new modeling data which, then is used to cycle through another loop in Figure 5.1.

When a MUMS modeling session is initiated, the user is requested to assess his or her expertise in performing the various structural analyses. The request and response are displayed in the window labeled "Analysis Types."

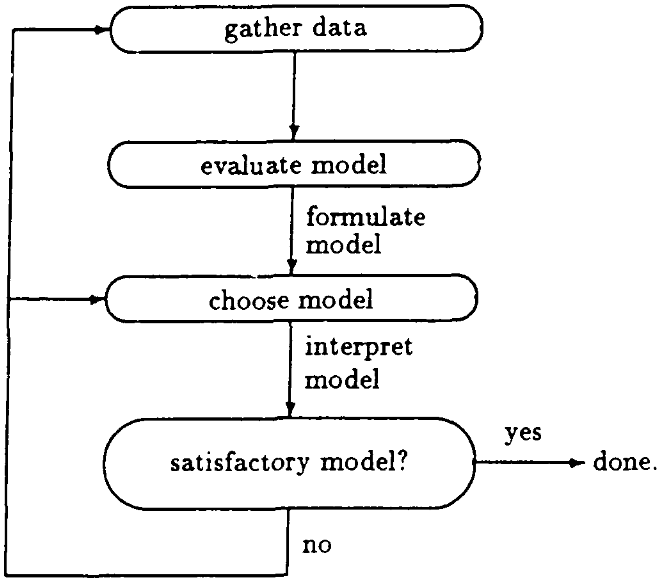


Figure 5.1 MUMS Control Structure

Analysis Types	Novice	Average	Expert
BACK-OF-ENVELOPE	□	■	□
ANALYTICAL-FORMULAS	□	□	■
QUALITATIVE-REASONING	□	□	■
FEM	■	□	□
TABLES	□	■	□
FIELD-TESTS	■	□	□
LAB-EXPERIMENTS	■	□	□

Here, the user chooses the *expertise* he or she possesses in each type of analysis. If the *expert* value is chosen for a particular analysis, the system will give more support for choosing that type of analysis in formulating the model. On the other hand, the choice *novice* will give less support. Once this is done, MUMS starts to ask questions about the structure by asking a question about focus, as seen in the next window.

What is the focus of the current analysis?
Preliminary-analysis
Final-design
Investigation
Cost-estimate
Feasibility-study

The box around *Final-design* indicates that the user has chosen the *Final-design* option. In other words, the ultimate goal of the user is a complete, final design of a structure. Now that the system has the purpose for the current structural model, it focuses on acquiring the data for the structure. According to the MUMS strategy rules, free (or cheap) actions which provide model data are preferred. Using this rule, MUMS then offers the user the following choice of data types to be entered:

Please choose something to ask or perform.
$\boxed{\text{b/a}}$
in-plane-loads
out-out-plane-loads
material
h/l

The user may choose to enter the values of any one of these five facts about the structure. In this example, the value of b/a is chosen first. Then the system asks for the value of b/a:

What is the value of b/a?
2.0

The user enters 2.0. Until the modeling data provide evidence to support the choice of an analysis, the user is prompted for more data (as in the following series of questions):

Please choose something to ask or perform.
in-plane-loads
out-out-plane-loads
material
$\boxed{\text{h/l}}$

The user chooses to enter the value of h/l:

What is the value of h/l?
25

And then the user is again prompted to choose a data type.

Please choose something to ask or perform.
in-plane-loads
out-out-plane-loads
$\boxed{\text{material}}$

The user chooses to enter the value of the structure's *material*:

Choose the material of the structure.
aluminum
timber
stone
$\boxed{\text{steel}}$
concrete
plastic

And then another data prompt follows.

Please choose something to ask or perform.
$\boxed{\text{in-plane-loads}}$
out-out-plane-loads

This time, the user enters the value of the *in-plane-loads*:

Choose the IN-PLANE loadings.
harmonic
$\boxed{\text{none}}$
pre-stress
random
shock
step
thermal
static

The value of the *out-of-plane-loads* is the only model data left to acquire:

Please choose something to ask or perform.
out-out-plane-loads

The user must enter the value *out-of-plane-loads*.

Choose the OUT-OF-PLANE loadings.
harmonic
none
pre-stress
random
shock
step
thermal
static

As the modeling data (such as the *material* or *h/l*) are entered by the user, MUMS propagates the effects of the new information through the inference network using the combination functions. As a result, new, intermediate conclusions about the model are generated. The status of MUMS conclusions are always visible to the user in the MU Output Window:

MU Output Window

[WM] Strength-of-assumption of DYNAMIC-MODEL is now STRONGLY-SUPPORTED
[WM] Strength-of-assumption of BUCKLING is now DISCONFIRMED
[WM] Trigger-level of CRACKING is now TRIGGERED
[WM] Strength-of-assumption of BRITTLE-FRACTURE is now SUPPORTED
[WM] Strength-of-assumption of FEM is now SUPPORTED
[WM] Strength-of-assumption of BACK-OF-ENVELOPE is now CONFLICTING
[WM] Trigger-level of BACK-OF-ENVELOPE is now TRIGGERED
[WM] Applicability of PERIOD-OF-INTEREST is now APPLICABLE
[WM] Strength-of-assumption of ANALYTICAL-FORMULAS is now STRONGLY-SUPPORTED
[WM] Trigger-level of SERVICEABILITY is now TRIGGERED
[WM] Trigger-level of STRENGTH is now TRIGGERED

From the MU Output Window, the various conclusions about the model are visible. At this stage of the model in this example, only intermediate conclusions can be made. This problem's data, for example, indicate that a BACK-OF-ENVELOPE calculation has been triggered. In other words, the BACK-OF-ENVELOPE calculation can provide data for the model, but at the present, it is unclear if the BACK-OF-ENVELOPE analysis is the best action to take.

The process of obtaining data from the user, propagating the data's effect in the inference network, and generating new conclusions is continued until an analysis hypothesis is confirmed. When an analysis hypothesis is confirmed, the system will display this fact in the MU Output Window (e.g., Strength-of-assumption of BACK-OF-ENVELOPE is CONFIRMED).

CHAPTER 6

Conclusions

MUMS is a partially implemented knowledge-based system for representing strategic choices in structural modeling. Based on the MU architecture for performing diagnostic reasoning, the MUMS system has an undeniable diagnostic disposition. And since structural modeling is not a purely diagnostic process, MUMS does have weaknesses in accurately representing structural modeling. However, the MU foundation does provide a strong base for flexible knowledge representation in structural modeling. This chapter describes some of the strengths and weaknesses of the MUMS system and suggests some future directions for research in the area.

6.1 MUMS: A Diagnostic System for Structural Modeling

The focus of the present research is the study of the application of MU, a task-level architecture for prospective diagnostic reasoning, to the structural modeling domain. The MU system was chosen for study because of the many similarities between prospective diagnosis and the process of structural modeling. Prospective diagnosis is concerned with selecting actions based on their potential consequences to the patient while structural modeling is concerned with selecting structural analyses based on their ability to provide information

to accurately represent the response of a structure. Furthermore, a prospective diagnosis may be complicated by conflicting goals. For example, a physician can perform a very painful or costly test as a means of evaluating a disease hypothesis when a less invasive test can be performed with little loss of diagnostic evidence. Whether the test is actually done depends on many factors including cost, time, and evidence gained. In structural modeling, an engineer can perform an FEM analysis on a particular structure (which may take hours) to find internal stresses when an analytical formula may be solved in a fraction of the time to provide more valuable information about the structure's response. Again, whether either analysis is performed depends on the time available, cost, model information gained and other features.

The MU system has many attributes which make it particularly useful as a structural modeling assistant. First, MU was developed to facilitate knowledge acquisition [8]. To this end, the knowledge in MU was made declarative not procedural. That is, knowledge in MU is represented in "localized packets" of knowledge whose "meaning" is explicit rather than in procedures to find that knowledge. Another product of MU's declarative knowledge base is the fact that explanations of actions are easily given in terms of the assumptions leading to the actions. This is accomplished simply by backtracking in the inference network to find the clusters which yield evidence for the action in question. Furthermore, maintenance is much easier since much of MU's knowledge is localized, that is, in that any one piece of knowledge does not affect large portions of the knowledge base. When knowledge does affect other knowledge in MU, it is done explicitly through the combination functions. In addition, a virtue of MU's declarative knowledge base is the fact that it is more easily acquired from structural modeling experts. This was very clear in the knowledge acquisition sessions. The *reasons* for performing modeling actions were more easily elicited from the expert than were *rules* for performing structural modeling actions. The structural modeling expert is

more likely to provide modeling knowledge in the form "I am using an analytical formula because the loading is uncertain and the analytical formula gives a more general solution." than in the form "If the loading is uncertain and an analytical formula gives a more general solution, then choose an analytical formula."

In the course of the MUMS research, it was found that MU's knowledge representation is consistent with the structural modeling task. Specifically, expert structural modelers use features in deciding what direction a structural modeling process should take. Surprisingly, many features used in diagnostic reasoning are the same as or similar to those used in structural modeling. Physicians use the **trigger** feature to bring to mind certain diseases when specific data are revealed. Analogously, engineers use the **trigger** feature to bring to mind a specific functional specification when certain modeling data are present. An example is the assumption of fatigue. When the number of loading cycles on a metal structure is large, the possibility of fatigue is quickly brought to mind. Moreover, as with physicians performing medical tests, structural modelers are not assumed to have equal access to tools for performing analyses or equal abilities in analyzing the results. Therefore, the modeler's environment and the ability of the modeler is a factor to be considered [5].

As expected, the MUMS research also revealed that the analogy between structural modeling and diagnosis is not perfect. In the analogy between medical diagnosis and structural modeling used here, the goal of a top-level goal of a diagnosis is the treatment of a disease using an assortment of medical tests and treatments. In structural modeling, the top-level goal is the formation "complete" model through the performance of various structural analyses. In structural modeling, the goal itself, the "complete" model, is often uncertain, whereas the clinical diagnostician's task is to find "what's wrong." In structural modeling, there is no "what's wrong."

6.2 Future Directions

A notable finding of the MUMS research is that the expert structural modeling knowledge of a single expert is likely to be limited to one or, at most, a few types of structural analyses. An engineer may be proficient at applying boundary element theory, for example, but not at applying the finite element method. An engineer may be expert at analytical formula applications for static problems, but not proficient at dynamic problems. An engineer may even be an expert in using one FEM package, but not another. Aside from the provisions for these differences — which must be included in a structural modeling assistant, the concerns of acquiring knowledge for such an assistant have to be considered. The pilot implementation of the MUMS system is based on the expert knowledge of one expert and so is skewed toward analytical formula analyses. To build a complete structural modeling assistant, knowledge acquisition must done with several structural modeling experts, and differences or conflicts in expert modeling strategies must be addressed and incorporated in the system.

Appendix A

Defined Features of MUMS

This appendix gives a few examples of the four different types of features defined in the MUMS system.

— Data Feature —

The **value** feature:

> **Feature-type:** Data
> **Value-type:** Unspecified
> **Value-range:** Unspecified
> **Value:** given by user or computed from model data

— Dynamic Features —

The **active-p** feature:
"An analysis which has been triggered but not ruled out is *active*."

> **Feature-type:** Dynamic
> **Value-type:** Ordinal
> **Value-range:** (active not-active)
> **Combination-function:** local to an analysis
> **Value:** to be inserted in the appropriate analysis type

The **applicability** feature:
"A question which is relevant to the current model. For example, a question about loading cycles is *applicable* if the structure has a dynamic load."

Feature-type: Dynamic
Value-type: Ordinal
Value-range: (applicable inapplicable)
Combination-function: local to a question
Value: to be inserted in the appropriate question frame

The **strength-of-assumption** feature:
"This feature is the value of the current degree of belief in the support for an object in the inference network."

Feature-type: Dynamic
Value-type: Ordinal
Value-range: (Disconfirmed Strongly-detracted Detracted
Conflicting Supported Strongly-supported Confirmed)
Combination-function: local to a cluster or analysis-type
Value: to be inserted in the appropriate cluster or analysis

The **potential-evidence** feature:
"The set of data which can potentially affect the level of support for an analysis. For example, an assumption of non-linear behavior will affect the support for a more complex analysis (increasing its level of support, in this case)"

Feature-type: Dynamic
Value-type: Ordinal
Value-range: (applicable inapplicable)
Combination-function: local to a frame
Value: to be inserted in the appropriate frame

— Static Features —

The **availability** feature:
"This feature is used in data and analysis nodes in the inference network to indicate whether the data or structural analysis is available to the modeler."

Feature-type: Static
Value-type: Ordinal
Value-range: (available not-available)
Combination-function: local to a frame
Value: to be inserted in the appropriate analysis frame

The **cost** feature:
"This feature is a measure of the cost for obtaining the data for a particular intermediate conclusion or analysis."

Feature-type: Dynamic
Value-type: Ordinal
Value-range: (free cheap low medium high very-high)
Combination-function: local to a frame
Value: to be inserted in the appropriate frame

The **number-of-dimensions** feature:
"This feature represents the number of dimensions for the current model (a beam, a plate, or a solid, for example)"

Feature-type: Dynamic
Value-type: Ordinal
Value-range: (1 2 3)
Combination-function: local to a analysis
Value: to be inserted in the appropriate analysis

— System Features —

The **network-dependents** feature:
"This feature keeps track of the nodes in the inference network which are dependent in any way on a particular object."

Feature-type: System
Value-type: a node in the inference network
Value-range: any inference network node
Combination-function: implementation-level
Value: to be inserted in the appropriate node by the knowledge engineer

APPENDIX B

Strategy Rules in MU/MUMS

— Rules that FOCUS —

This will focus upon actions which have a possibility of leading to a conclusion e.g. an intermediate conclusion to confirm a cluster.

```
Focus rule 1: Focus on Conclusive Evidence
If:
      (IS (DIFFERENTIAL) :NONEMPTY)
      (IN ?ACTION
       (POTENTIAL-EVIDENCE
        DIFFERENTIAL))
Then:
      (PROPOSE ?ACTION
       GATHER-EVIDENCE-FOR-DIFFERENTIAL)
```

This rule will choose questions (from the set of intial data-gathering questions) to ask when MUMS cannot find an appropriate analysis using the data of the current model.

```
Focus rule 2: Ask Identifying Questions
If:
      (IS (DIFFERENTIAL) :EMPTY)
      (IN ?ACTION
       (MEMBERS-OF INITIAL-QUESTIONS))
Then:
      (PROPOSE ?ACTION COMPLETE-MODEL)
```

This rule will choose questions (from the set of general data questions) to ask when MUMS cannot find an appropriate analysis using the data of the current model.

```
Focus rule 3: Ask Model Data Questions
If:
      (IS (DIFFERENTIAL) :EMPTY)
      (IN ?ACTION
       (MEMBERS-OF GENERAL-QUESTIONS))
Then:
      (PROPOSE ?ACTION COMPLETE-MODEL)
```

This rule will end the modeling session when MUMS concludes that a certain analysis should be performed.

```
Focus rule 4: Halt on Confirmed Hypothesis
If:
      (IN ?HYPO (DIFFERENTIAL))
      (IS (STRENGTH-OF-ASSUMPTION ?HYPO)
       CONFIRMED)
Then:
      (PROPOSE HALT HALT ?HYPO
  is confirmed.)
```

— Rules that FILTER —

This rule will remove from consideration any action which has already been performed (i.e. so that a question will not be asked twice with the current model).

```
Filter rule 1: Filter Executed Actions
If:
      (IS (EXECUTED? ?ACTION) YES)
Then:
      (FILTER ?ACTION already executed)
```

This rule will remove from consideration any question which is, in any way, not applicable in the present situation.

```
Filter rule 2: Filter Inapplicable Questions
If:
      (IS (APPLICABILITY ?ACTION)
       INAPPLICABLE)
Then: (FILTER ?ACTION needs prerequisites)
```

— Rules that PREFER —

This rule will give a preference to performing cheap actions which potentially trigger a hypothesis.

```
Prefer rule 1: Prefer Cheap Triggering Data
If:
      (IN COMPLETE-MODEL (CURRENT-GOALS))
      (IS (POTENTIALLY-TRIGGERED ?ACTION)
       :NONEMPTY)
      (<= (COST ?ACTION) CHEAP)
Then:
      (PREFER ?ACTION)
```

This rule will give a preference to those actions which are both cheap and have a possibility of leading directly to the choice of a structural analysis.

```
Prefer rule 2: Prefer Cheap Conclusive Evidence
If:
      (IN
       GATHER-EVIDENCE-FOR-DIFFERENTIAL
       (VALUE CURRENT-GOALS))
      (IN ?ACTION
       (POTENTIALLY-CONCLUSIVE-EVIDENCE
        DIFFERENTIAL))
      (<= (COST ?ACTION) CHEAP)
Then:
      (PREFER ?ACTION)
```

This rule will give a preference to those actions which are both free and have a possibility of leading indirectly to the choice of a structural analysis.

```
Prefer rule 3: Prefer Free Conclusive Evidence
If:
      (IN
       GATHER-EVIDENCE-FOR-DIFFERENTIAL
       (VALUE CURRENT-GOALS))
      (<= (COST ?ACTION) FREE)
      (IN ?ACTION
       (POTENTIAL-EVIDENCE
        DIFFERENTIAL))
Then:
      (PREFER ?ACTION)
```

This rule will give a preference to free actions.

```
Prefer rule 4: Prefer Free Evidence
If:
     (<= (COST ?ACTION) FREE)
Then:
     (PREFER ?ACTION (COST ?ACTION))
```

This rule will give a preference to those actions which have a possibility of leading directly to the choice of a structural analysis regardless of the actions cost.

```
Prefer rule 5: Prefer Conclusive Evidence
If:
     (IN
      GATHER-EVIDENCE-FOR-DIFFERENTIAL
      (CURRENT-GOALS))
     (IN ?ACTION
      (POTENTIALLY-CONCLUSIVE-EVIDENCE
       DIFFERENTIAL))
Then:
     (PREFER ?ACTION conclusive evidence)
```

This rule gives a preference to actions which are cheap.

```
Prefer rule 6: Prefer Cheap Evidence
If:
     (<= (COST ?ACTION) CHEAP)
Then:
     (PREFER ?ACTION (COST ?ACTION))
```

This rule will give a preference to performing actions which potentially trigger a hypothesis regardless of the action's cost.

```
Prefer rule 7: Prefer Triggering Data
If:
     (IN COMPLETE-MODEL (CURRENT-GOALS))
     (IS (POTENTIALLY-TRIGGERED ?ACTION)
      :KNOWN)
Then:
     (PREFER ?ACTION)
```

Bibliography

[1] J. Bennett, L. Creary, R. Englemore, and R. Melosh. *SACON:A Knowledge-Based Consultant for Structural Analysis.* Technical Report STAN-CS-78-699, Department of Computer Science, Stanford University, 1978.

[2] P. Cohen and T. Gruber. *Reasoning About uncertainty: a knowledge representation perspective.* Technical Report 85-24, Department of Computer and Information Sciences, University of Massachusetts, 1985.

[3] P. R. Cohen, M. Greenberg, and J. DeLisio. Mu: a development environment for prospective reasoning systems. In *Proceedings of AAAI-87 Sixth National Conference on Artificial Intelligence*, July 1987.

[4] P. R. Cohen, M. L. Greenberg, D. M. Hart, and A. E. Howe. *Trial by Fire: Understanding the Design Requirements for Agents in Complex Environments.* Technical Report TR 89-61, COINS Dept., University of Massachusetts, 1989.

[5] J. R. Dixon, C. D. Jones, S. C. Luby, E. C. Libardi, and E. H. Nielson. Knowledge representation in mechanical design. *SAE Technical Paper Series*, 11–20, 1986.

[6] C. L. Dym, R. P. Henchey, E. A. Delis, and S. Gonick. A knowledge-based system for automated architectural code checking. *Computer-Aided Design*, 20:137–145, 1988.

[7] T. Gruber. *The Acquisition of Strategic Knowledge.* PhD thesis, University of Massachusetts, 1989.

[8] T. R. Gruber and P. R. Cohen. Design for acquisition: principles of knowledge system design to facilitate knowledge acquisition. *IJMMS*, 26(2):143–159, 1987.

[9] C. T. Kitzmiller and J. S. Kowalik. Workshop report: coupling symbolic and numeric computing in knowledge-based systems. *AI Magazine*, 8:85–90, 1987.

[10] V. R. Lesser, J. Pavlin, and E. Durfee. Approximate processing in real-time problem solving. *AI Magazine*, 9:49–61, 1988.

[11] S. Mittal and C. L. Dym. Knowledge acquisition from multiple experts. *AI Magazine*, 6:32–36, 1985.

[12] E. Rich. *Artificial Intelligence*. McGraw-Hill, New York, 1983.

[13] I. H. Shames and C. L. Dym. *Energy and Finite Element Methods in Structural Mechanics*. Hemisphere Publishing Corp., 1985.

[14] E. H. Shortliffe and B. G. Buchanan. A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23:351–379, 1975.

[15] G. M. Turkiyyah and S. J. Fenves. Knowledge-based analysis of structural systems. *Proceedings of the Second International Conference on Artificial Intelligence in Engineering*, II:273–284, 1987.

[16] J. R. Zumsteg and D. L. Flaggs. Knowledge-based analysis and design systems for aerospace structures. In C. L. Dym, editor, *Applications of Knowledge-Based Systems to Engineering Analysis and Design*, pages 67–80, New York: American Society of Mechanical Engineers, 1985.